Web Project

Personal Project
2024.

Portfolio

Index

Part.1

프로젝트 요약

01 프로젝트 주제

여행 목적지별 목록을 통해, 고객들에게 다양한 여행 패키지를 제시한다.

02 선정 배경

여행이라는 단어만으로도 설렘을 느낄 수 있듯 인생에서 잊지 못 할 추억을 만들어 주는 ‘ 여행 '이란 단어를 한 페이지에 담아 내고 싶어 선정하게 되었습니다.

03 기획 의도

고객이 여행을 예약하는 과정을 간소화 할 수 있는 방법을 찾고, 향후 각종 여행지에 대한 유저간 커뮤니티를 만들어 정보 교류 채널이 되기 위해 페이지를 기획하게 되었습니다.

Part.2

프로젝트 수행 방법

BACK

Java
Thymeleaf

FRONT

HTML
CSS
Java Script

DB

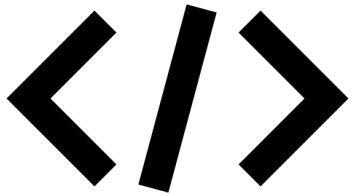
Oracle

TOOL

Spring Boot
Visual Code

Part.3

프로젝트 수행 결과



01

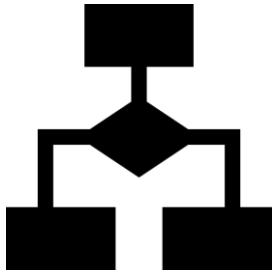
CODE

웹 프로젝트 JAVA 클래스의
역할과 전체적인 코드 구조 VIEW

02

PAGE VIEW

웹 사이트 페이지별 VIEW 구성



03

USE CASE DIAGRAM

웹 페이지 사용자별 전체 페이지 구조

Code

```java
@Controller

public class cont {

    @Autowired
    public Mapping_data Mapp;

    @GetMapping("/")
    public String index()
    {


        return "index.html";
    }

    @GetMapping("/joinus")
    public String joinus( )
    {

        return "joinus.html";
    }

    @PostMapping("/joinus")
    public String member_join(@RequestParam String join_id,@RequestParam String join_pw,@RequestParam String join_pw1,@RequestParam String join_name,
                              @RequestParam String join_cell1,@RequestParam String join_cell2,@RequestParam String join_addr)

    {
            DTO_builder data = new DTO_builder();
                data.Setid(join_id)
                .Setpw(join_pw1)
                .Setpw1(join_pw1)
                .Setname(join_name)
                .Setcell1(join_cell1)
                .Setcell2(join_cell2)
                .Setaddr(join_addr)
                .build();

            Mapp.insert_data(join_id, join_pw, join_pw1, join_name, join_cell1, join_cell2, join_addr);
            return "redirect:/login_page";
    }
```

```javascript
    const idRegExp = /^[a-zA-Z0-9]{4,12}$/;
    const passRegExp = /^[a-zA-Z0-9]{4,12}$/;
    const nameRegExp =/^[ㄱ-ㅎ가-힣]{1,6}$/;

    function checkAll()
    {
        if(!checkId()){ return false;}
        else if(!Pass()){return false;}
        else if(!checkPass()){return false;}
        else if(!checkName()){return false;}
        else if(!checkSnum()){return false;}
        alert('회원가입 성공');
        window.location.href ="index";
        document.forms[0].submit();
        return true;
     }
     function checkId(){
        const box = document.getElementById('input_id');
        if(!idRegExp.test(box.value)){
            alert('아이디를 정확하게 적어주세요.');
            box.value = "";
            box.focus();
            return false;
        }
        return true;
    }

     function Pass(){
        const box = document.getElementById('input_pw');
        if(!idRegExp.test(box.value)){
            alert('비밀번호를 잘 적어주세요.');
            box.value = "";
            box.focus();
            return false;
        }else if(box.value == document.getElementById('input_id').value){
            alert('비밀번호는 아이디와 같을 수 없습니다.')
            box.value = "";
            box.focus();
            return false;
        }
```

MAIN PAGE - JOINUS

회원가입 페이지 구현
아이디 , 패스워드, 이름, 주민번호, 주소 데이터를 입력하면
회원가입이 가능하다.

HTML :
· Form을 통한 회원 데이터 입력
· Function(onclick) 을 통한 입력데이터 확인 후 데이터 전송
· 정규식을 이용하여 특정 텍스트만 허용

JAVA :
·Mapper
SQL구문을 적용한 데이터베이스 내 테이블 데이터 접근

·Controller
Form으로 전송 된 회원가입 데이터 기반 SQL구문 실행
회원가입 완료 시 로그인 페이지로 이동

Code

```java
@Configuration
@EnableWebSecurity
public class secure {

    @Bean
    protected PasswordEncoder passwordEncoder()
    {
        return new BCryptPasswordEncoder();
    }


    @Bean
    public SecurityFilterChain filterChain(HttpSecurity http ) throws Exception
    {

        http.csrf(csrf->csrf.disable());

        http.authorizeHttpRequests(authz->authz.requestMatchers("/login_page","/admin/system","/notice").authenticated());

        http.authorizeHttpRequests
                (authz->authz.requestMatchers("/**","tempalates","static").permitAll()
                .requestMatchers("/notice").hasAnyRole("ADMIN","USER")
                .requestMatchers("/admin/system").hasRole("ADMIN").anyRequest().permitAll());

        http.formLogin
                (formlogin -> formlogin.loginPage("/login_page").loginProcessingUrl("/login_page")
                .failureForwardUrl("/loginerror?login_error=1")
                .usernameParameter("username")
                .passwordParameter("password")
                .defaultSuccessUrl("/",true).failureUrl("/login_page").permitAll());

        http.logout
                (logout->logout.logoutUrl("/logout").logoutSuccessUrl("/login_page")
                .addLogoutHandler((request,response,authentication)->
                {HttpSession session = request.getSession();
                session.invalidate();})
                .logoutSuccessHandler((request,response,authentication)->response.sendRedirect("/login_page")).deleteCookies("JESSIONID","cookie"));

            return http.build();
    }
```

```java
    @Autowired
    private Mapping_data mapper;

    @Bean
    public InMemoryUserDetailsManager InMemoryUserDetailsManager() throws Exception
    {
        return new InMemoryUserDetailsManager();
    }


    @Primary
    @Bean
    public AuthenticationManagerBuilder configure(AuthenticationManagerBuilder auth) throws Exception
    {
        List<dto> li =mapper.select();
        Iterator<dto> it = li.iterator();
        while(it.hasNext())
        {
            for(dto i:li)
            {
                it.next();
                if(i.getid().equals("admin"))
                {
                    auth.inMemoryAuthentication().withUser(i.getid()).password(passwordEncoder().encode(i.getpw())).roles("ADMIN");
                }
                else {
                    auth.inMemoryAuthentication().withUser(i.getid()).password(passwordEncoder().encode(i.getpw())).roles("USER");
                }
            }
        }
        return auth;
    }

}
```

MAIN PAGE – LOGIN & LOGOUT

로그인 및 로그아웃 페이지 구현
ID,PASWWORD 입력 시 권한에 따른 사용자 접근 권한이 주어지며 데이터베이스 내 ID,PASSWORD를 매칭하여 로그인이 가능하다.

HTML :
· Form을 통한 ID,PASSWORD 데이터 입력
· 비회원 페이지 접속 시 LOGIN 활성화
· 회원 로그인 시 LOGOUT,MYPAGE 활성화

JAVA :
·Spring Security
Custom Form Login 필터 적용 로그인 페이지 연결
로그인 데이터 매칭 실패 시 , 로그인 페이지로 재 연결
데이터베이스 데이터 유저 ID 기반 유저 권한 설정
페이지별 User와 Admin 접근 권한 설정
사용자 패스워드 부호화 작업을 통한 인코딩 작업
로그아웃 시 세션 및 쿠키데이터 삭제

Code

```java
package HOMEPAGE.Map;

import java.util.List;
import org.apache.ibatis.annotations.Insert;
import org.apache.ibatis.annotations.Mapper;
import org.apache.ibatis.annotations.Select;
import org.apache.ibatis.annotations.Update;
import HOMEPAGE.DTO.dto;


@Mapper
public interface Mapping_data {

    @Insert("insert into joinform values(#{join_id},#{join_pw},#{join_pw1},#{join_name},#{join_cell1},#{join_cell2},#{join_addr})")
    public void insert_data(String join_id ,String join_pw, String join_pw1,String join_name, String join_cell1 , String join_cell2 ,String join_addr );

    @Select("select * from joinform")
    public List<dto> select();

    @Insert("insert into notice values(#{id},#{title},#{email},#{data})") //문의사항
    public void insert_board(String id,String title ,String email, String data);

    @Update("update joinform set pw='${join_pw}', pw2='${join_pw}' where id='${join_id}'")
    public void updata_pw(String join_pw, String join_id);


}
```

```java
    @GetMapping("/information")
    public String information()
    {
        return "information.html";
    }

    @PostMapping("/notice")
    public String data(@RequestParam String id ,@RequestParam String title, @RequestParam String email, @RequestParam String data)
    {

        Mapp.insert_board(id, title,email,data);
        return "notice.html";
    }


    @PostMapping("/mypage")
    public String mypage(@RequestParam String join_pw, @RequestParam String join_id) {

        if(join_id != null)
        {
            Mapp.updata_pw(join_pw, join_id);
            return "redirect:/";
        }

        return "/user/mypage.html";
    }


    @GetMapping("/notice")
    public String notice()
```

MAIN PAGE - MYPAGE

로그인 시 MYPAGE 메뉴 생성
로그인 회원만 MYPAGE 노출이 되며 , 데이터베이스 ID,PW 데이터를 기반으로 기존 패스워드 수정이 가능하다.

HTML :
· INPUT 을 통한 ID,PASSWORD 데이터 입력
· Thymeleaf security 인증 정보 매칭 후 메뉴버튼 활성화

JAVA :
·Mapper
SQL구문을 적용한 데이터베이스 내 PASSWORD 업데이트

·Controller
Form으로 전송 된 ID,PW 기반 SQL구문 실행
패스워드 수정 완료 시 메인 페이지로 이동

Code

```java
package HOMEPAGE.DTO;
import lombok.Data;

@Data
public class dto {

    private String join_id,join_pw, join_pw1, join_name, join_cell1, join_cell2, join_addr;

    public dto (String join_id,String join_pw,String join_pw1,String join_name,String join_cell1,String join_cell2, String join_addr )
    {
        this.join_id=join_id;
        this.join_pw=join_pw;
        this.join_pw1=join_pw1;
        this.join_name=join_name;
        this.join_cell1=join_cell1;
        this.join_cell2=join_cell2;
        this.join_addr=join_addr;
    }
    public String getid()   {   return join_id;     }
    public String getpw()   {   return join_pw;     }
    public String getpw1()  {   return join_pw1;    }
    public String getname() {   return join_name;   }
    public String getcell1(){   return join_cell1;  }
    public String getcell2(){   return join_cell2;  }
    public String getaddr() {   return join_addr;   }

    public dto build()
    {   return new dto(join_id,join_pw,join_pw1,join_name,join_cell1,join_cell2,join_addr);     }
```

```java
import lombok.Data;

@Data
public class DTO_builder {

    private String join_id,join_pw, join_pw1, join_name, join_cell1, join_cell2, join_addr;

    public DTO_builder Setid(String join_id) {
        this.join_id = join_id; return this; }

    public DTO_builder Setpw(String join_pw) {
        this.join_pw = join_pw; return this; }

    public DTO_builder Setpw1(String join_pw1) {
        this.join_pw1 = join_pw1; return this; }

    public DTO_builder Setname(String join_name) {
        this.join_name = join_name; return this; }

    public DTO_builder Setcell1(String join_cell1) {
        this.join_cell1 = join_cell1; return this; }

    public DTO_builder Setcell2(String join_cell2) {
        this.join_cell2 = join_cell2; return this; }

        public DTO_builder Setaddr(String join_addr) {
        this.join_addr = join_addr; return this; }

    public dto build()  {   return new dto(join_id,join_pw,join_pw1,join_name, join_cell1, join_cell2, join_addr);  }
```

CUSTOMER DATA

데이터 가독성을 위한 Builder Pattern 생성

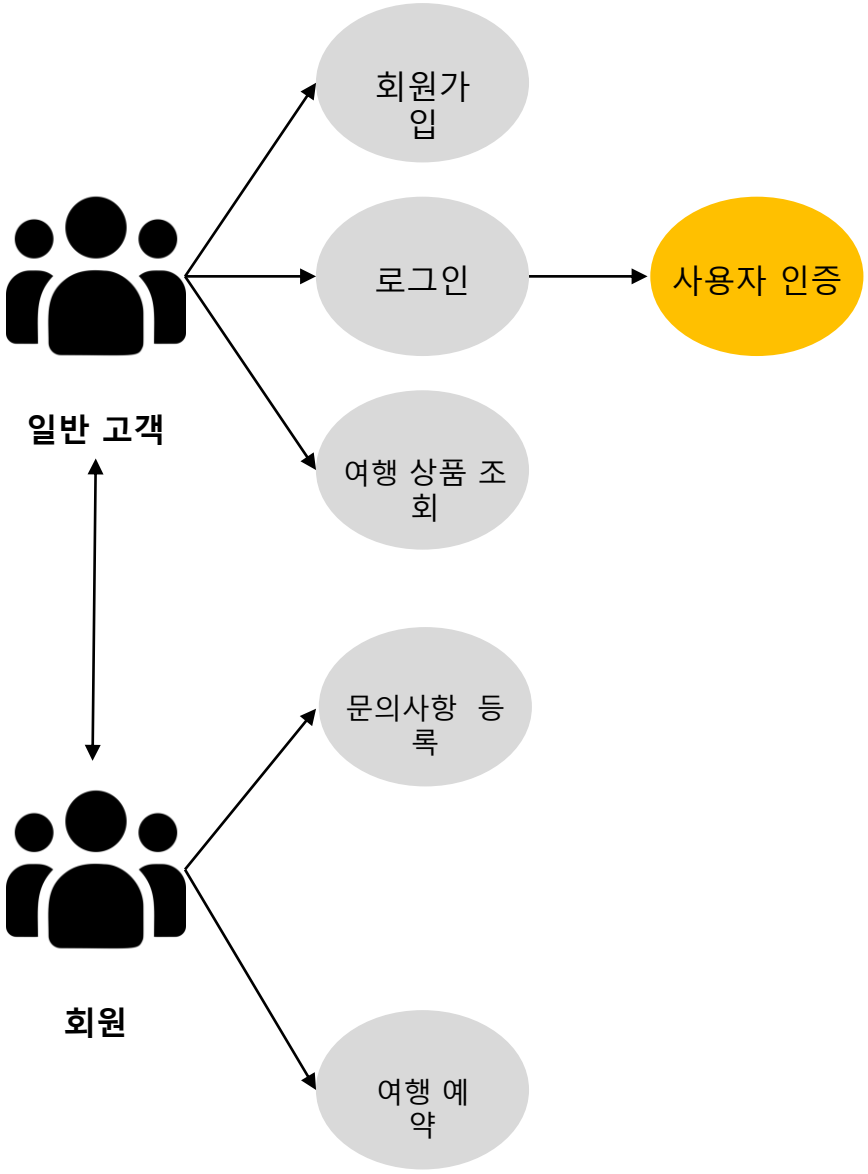
JAVA :

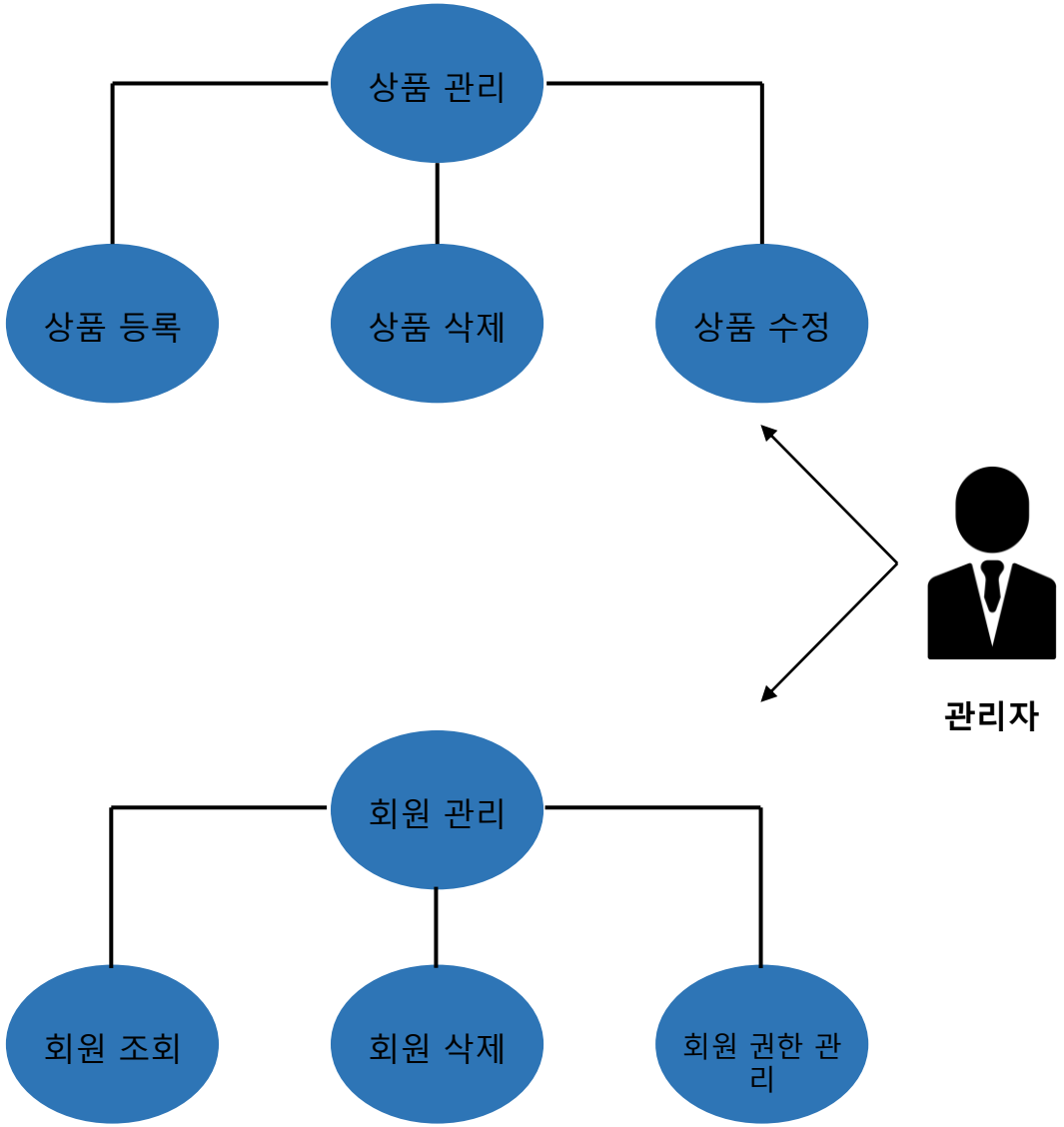
·Builder Pattern을 활용하여 데이터의 식별성과 보안성 증대

·회원가입 데이터와 같은 객체의 일관성 주입

Part.3

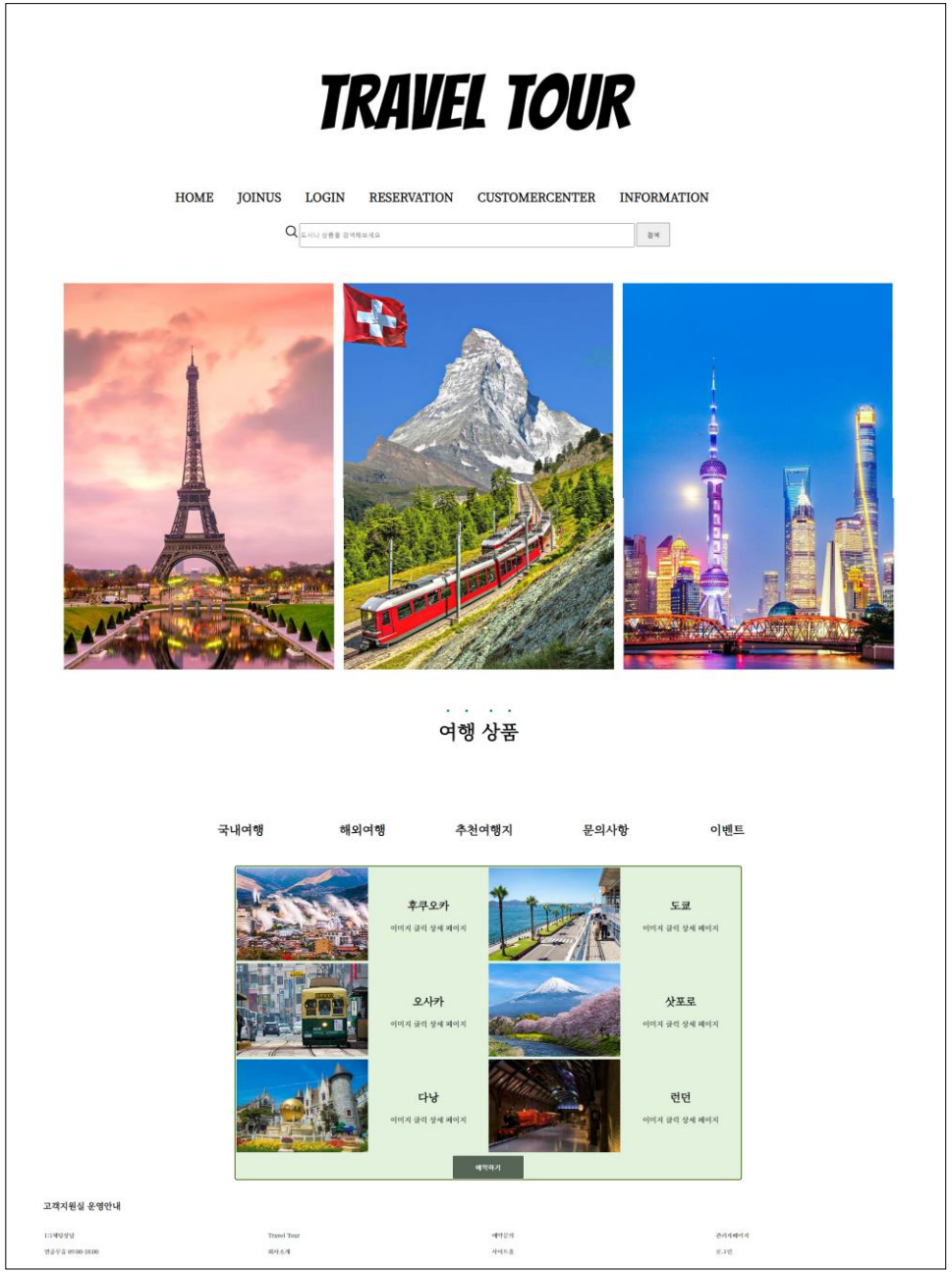
프로젝트 수행 결과

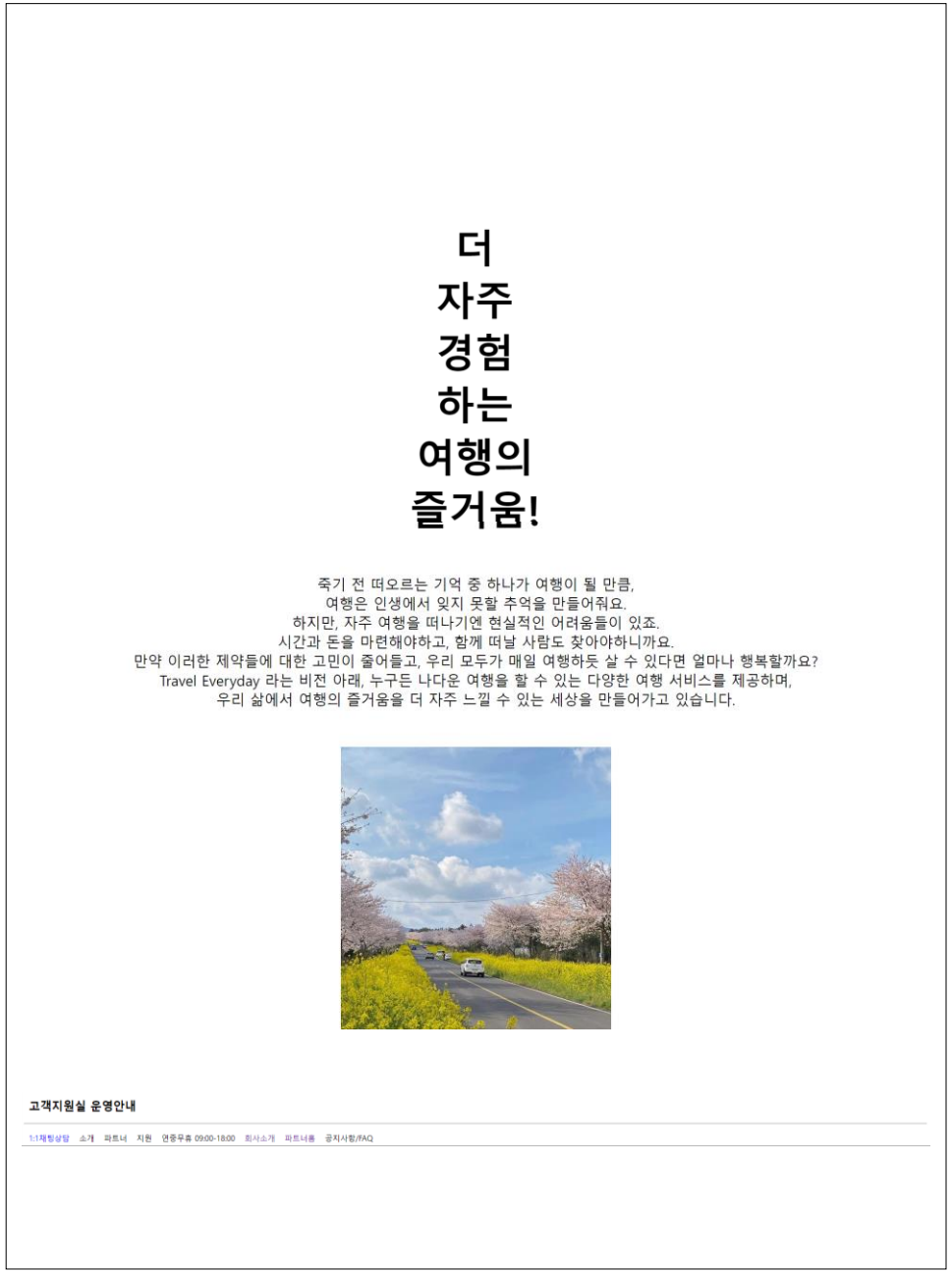




Part.4

프로젝트 페이지 뷰





Part.4

프로젝트 페이지 뷰

FUKUOKA 여행 안내

여행 시작

보다 더 자세한 일정.

출발 만나는 장소 8:10~ 8:25

만나는 장소는 일본 후쿠오카 하카타역 오리엔탈
호텔 로손 편의점 앞입니다.
후쿠오카 하카타역 오리엔탈 호텔 로손 편의점 앞에서
여행 당일 오전 8시 10분부터 오전 8시 25분까지
여행자분과 만남을 시작합니다.
당일 담당 여행 안내원님께서 여행 한 그룻 깃발을
들고 계실 예정입니다.

여행 상품

무료 사진 촬영 포함 후쿠오카 버스투어, 다자이후 유후인 유후다케 벳푸

1인당 49000원

무료 사진 촬영 포함 후쿠오카 버스투어, 나가사키 하우스텐보스

1인당 39800원

무료 사진 촬영 포함 후쿠오카 버스투어, 시모노세키 모지코 고쿠라

1인당 39800원

오사카 출발 나라 고베 버스 투어 여행, 그리고 무료사진 촬영

1인당 50000원

일본 관서지방 오사카 출발 교토 버스투어 여행

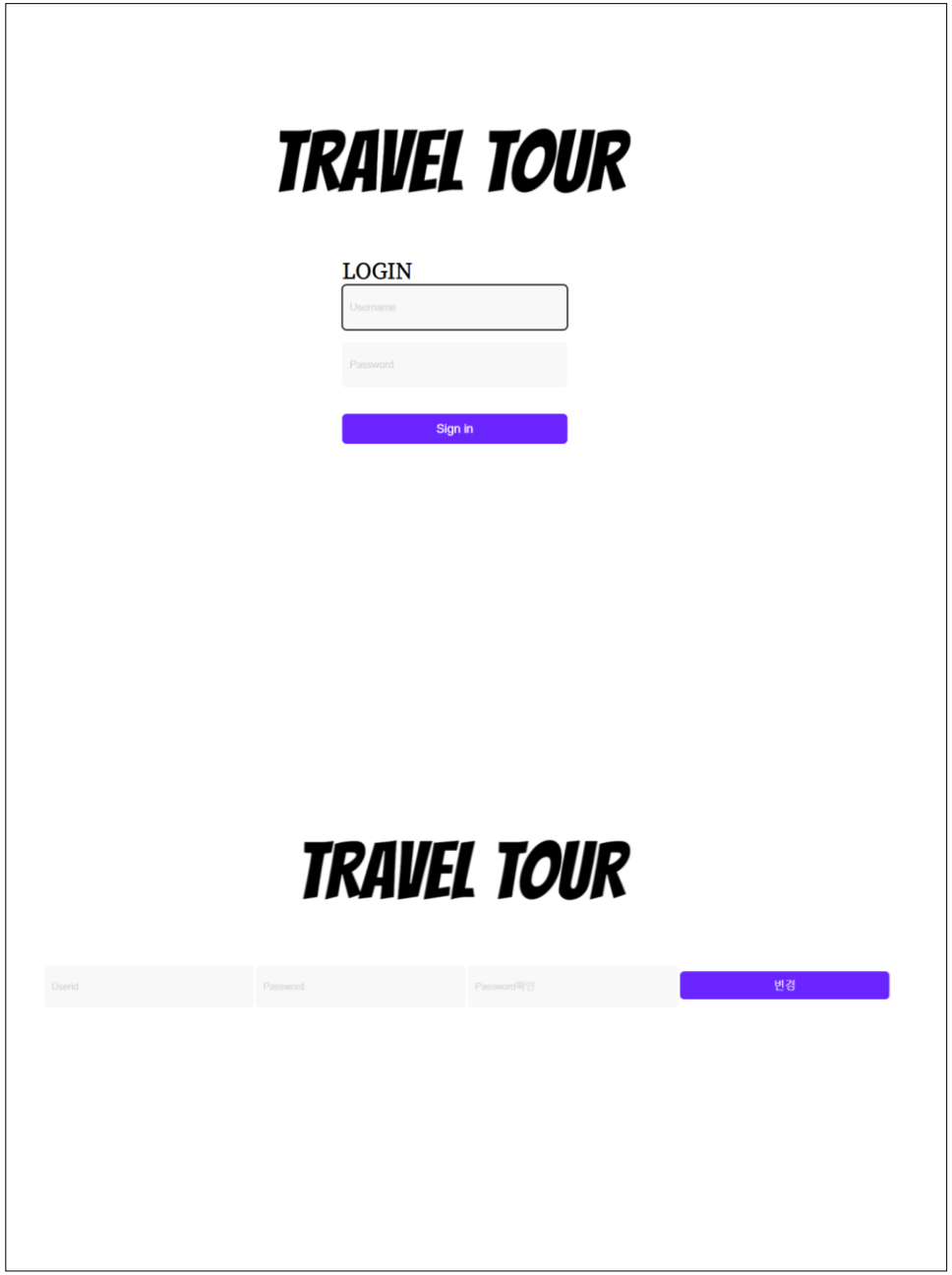
1인당 55000원

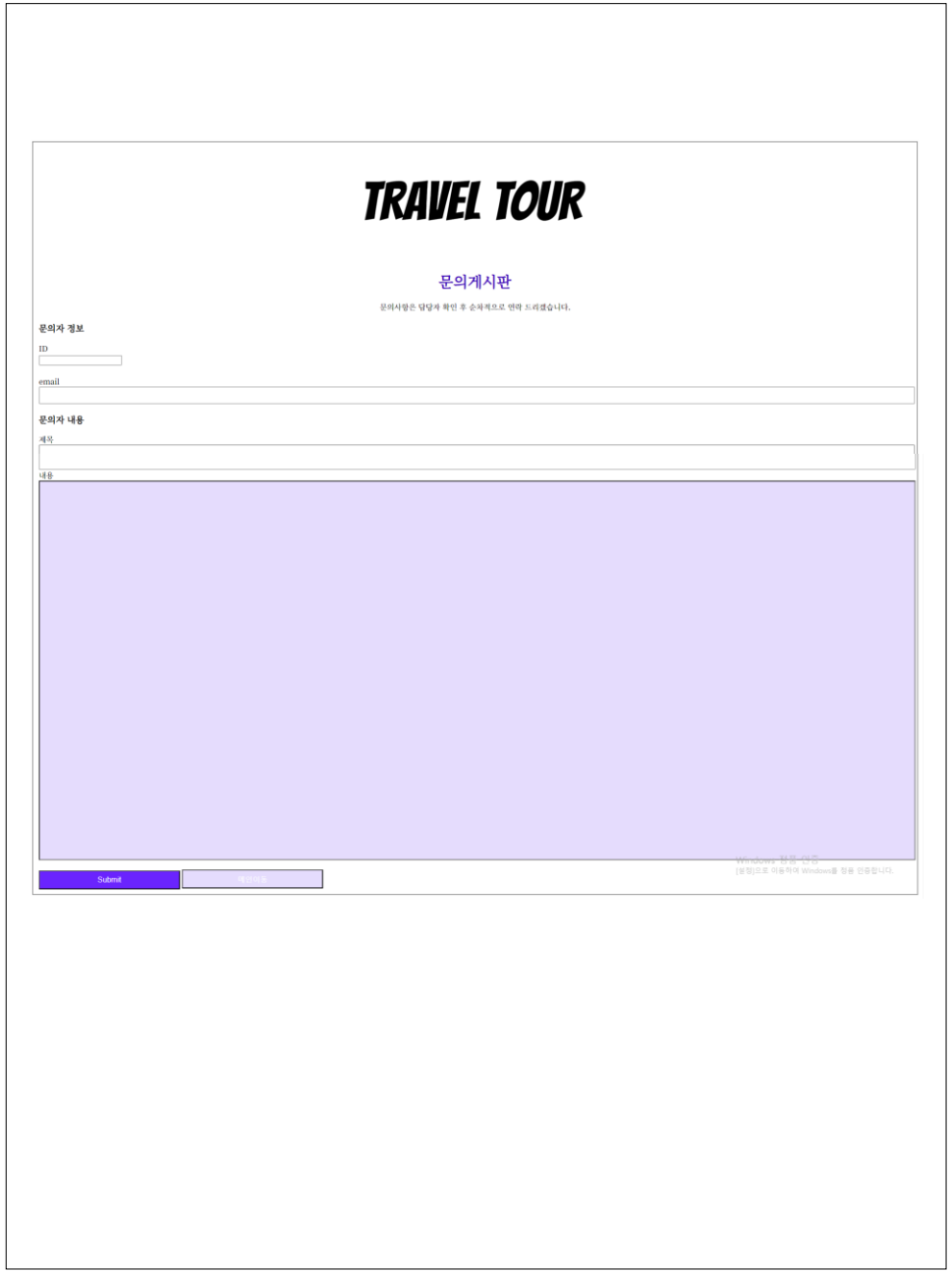
일본 관서지방 오사카 출발 교토 버스투어 여행

1인당 100000원

Part.4

프로젝트 페이지 뷰





Part.5

프로젝트 AAR

After Action Review

01 목표

얻고자 한 것 :

웹페이지에 구축에 사용되는 언어와 Tool 사용법을 숙지하고, 전체적인 구조에 대한 이해

02 결과

실제로 얻은 것 :

사전 작업 계획서 수립의 중요성과 Security 구조에 대한 이해도

03 원인 분석

차이와 그 원인 :

웹페이지의 전체 레이아웃 계획에서 어떤 기능을 사용할 것인지 구체적으로 작성하지 못 해 기능 선택에 대한 시간 할당이 많이 소요되었고, 세션과 쿠키 데이터에 대한 이해도 부족으로 보안관련 코드 구축에 어려움이 있었음.

04 피드백 종합

피드백 교훈:

프로그램 설계자는 사전 계획 수립을 토대로 작업 시간 스케줄링과 구축 순서가 중요하다는 것을 깨달았고, 더 많은 기능을 담지 못한 아쉬움이 있었다.

Personal Project
2024.

End

Git hub :
https://github.com/PARKYISEUL8153/Travel_Project.git